

Rapport Qualité de Donnée

Groupe :

ALI Houssam

DEBIANE Mohamed Ramdane

HAMMAD Amir

I. Mappings

A. Spécification des mappings

Après avoir analysé le schéma des tables contenu dans les sources et le schéma cible, nous avons déterminé les requêtes qui permettent de calculer les tables cibles à partir des sources en utilisant le modèle Global-As-View.

Nous avons déterminé les mappings suivants :

- Table intermédiaire :

Pour faciliter le calcul d'agrégat que représente certaines colonnes de la table cible, nous avons choisi de stocker les résultats de jointures successives dans une table intermédiaire sur laquelle seront effectuées des restrictions sur le code postal et la localisation afin d'effectuer les

L'expression de calcul de la table intermédiaire est la suivante :

$$\text{tmp} = (\text{S1.Station} \bowtie \text{S1.Mesure}) \cup (\text{S2.Station} \cup \text{S2.Mesure}) \bowtie \text{Polluant} \bowtie (\text{S4.Mesure} \cup \text{S5.Mesure})$$

Une fois la table temporaire (tmp) définie, le calcul des attributs : Taux_moyen_jour, NBS(Nombre de stations), NBC (Nombre de Capteurs) se fait à l'aide des expressions suivantes.

$$\text{Taux_Moyen_Jour} = (\text{SUMS} + \text{SUMC}) / (\text{NBC} + \text{NBS}) _(\text{tmp})$$
$$\text{NBS} = \text{COUNT}(\text{DISTINCT id_Station}) \sigma _ \text{localisation_}(\text{tmp})$$
$$\text{NBC} = \text{COUNT}(\text{DISTINCT ID_Capteur}) \sigma _ \text{localisation_}(\text{tmp})$$

Ici SUMS et SUMC représentent les sommes respectives de toutes les stations et de tous les capteurs.

Le statut se calcule par comparaison entre le taux_moyen_relevé une fois calculé et le seuil toléré qui est un attribut de la table Polluant

Remarque :

Les trois tables peuvent être peuplées à l'aide des mêmes requêtes, car il suffit de modifier le critère de restriction avec le code postal ou le nom du département correspondant.

- Table Station :

Le mapping des stations s'écrit en fonction des sources S1 et S2 qui contiennent les tables Station de schéma similaire. Ces sources étant disponibles dans la table tmp, une simple projection suffit à les récupérer.

Cible.Station = TT Id_Station, Adresse,Tel, Contact_Mail(tmp)

Dans la table Cible.Station, l'attribut adresse est représenté par la concaténation des attributs : Num + Rue + Ville + Code_Postal

.

- Tables Cibles :

Les tables Paris, Hauts_de_Seine, Yvelines ont quant à elles des schémas très similaires qui reposent sur le calcul de certains agrégats comme la somme du nombre de stations et de capteurs mobiles, ainsi que du taux moyen d'un polluant sur durée en jours donnée.

Nous détaillons l'implémentation de ces tables dans la section suivante.

B. Implémentation des mappings

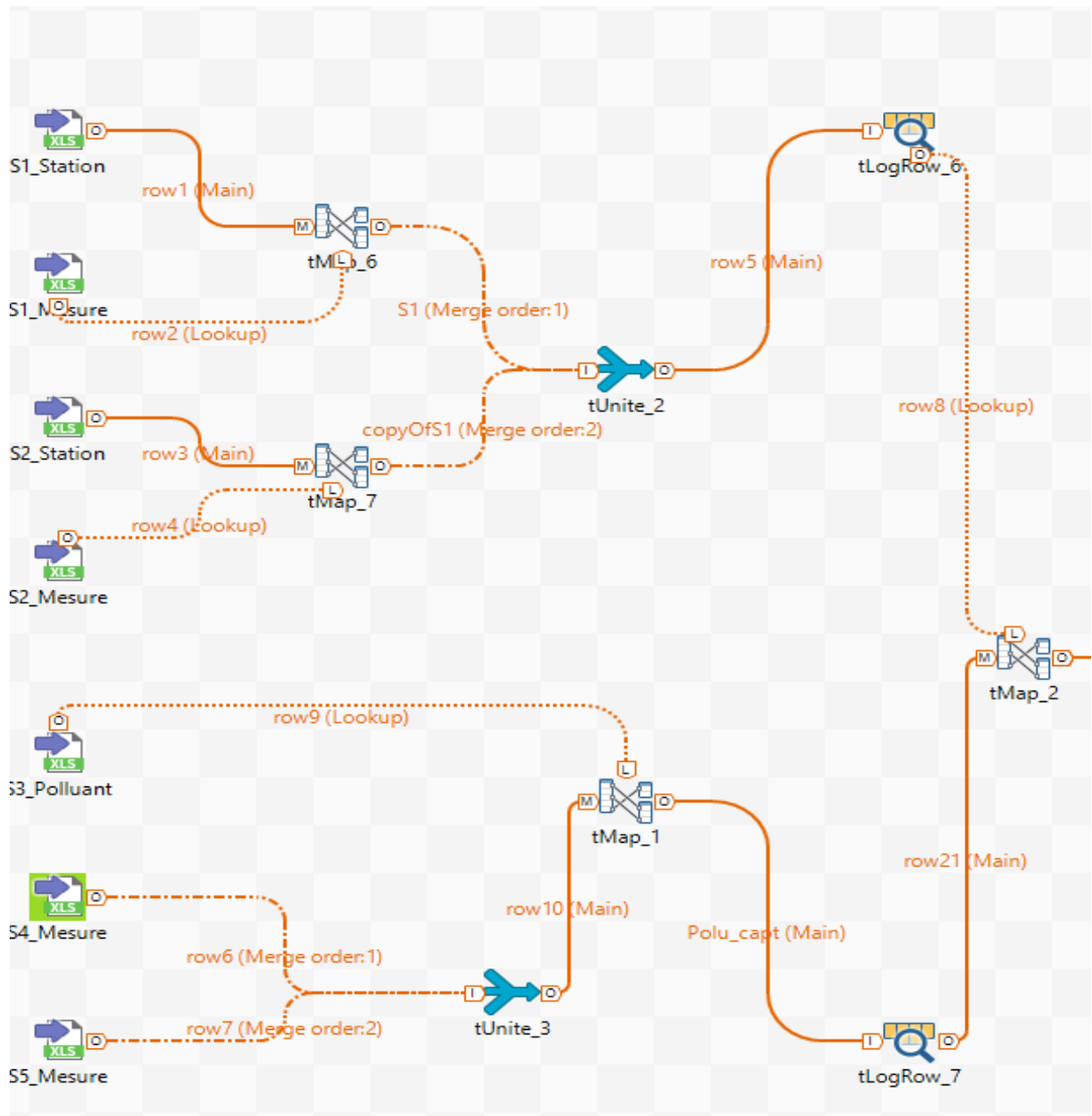
Nous avons utilisé les éléments mis à notre disposition par talend pour créer la table temporaire de la façon suivante :

- Création de la table tmp :

Le schéma ci-dessous matérialise l'expression de la table tmp :

- tMap permet de faire la jointure entre Station et Mesure de la source de donnée 1 et 2 respectivement appelé tMap_6 et tMap_7.

- Le composant tUnite fait l'union des données de différentes sources en se basant sur un schéma commun soit les sous ensembles des flux de sortie tMap_6 et tMap_7 sur notre schéma.
- Le composant tUnite3 fait l'union des différentes mesures de capteurs qui seront ensuite joints avec tMap_1 avec les polluants.
- Le composant principal tMap1 fait la jointure entre les mesures des stations et des capteurs sur le critère (ID_polluant et date). Nous voulons les mesures faites par un capteur et/ou une station au même moment sur un même polluant.



- Création de la table Station :

Le schéma ci-dessous matérialise le mapping décrit précédemment :

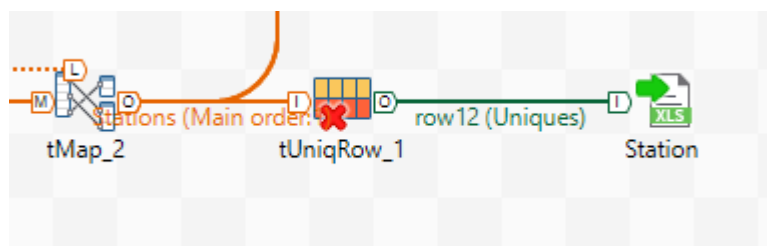
De plus grâce à tMap nous pouvons effectuer une concaténation des attributs num, Rue, Ville et Code_postal.

À partir de la table temporaire nous gardons les éléments suivants :

Stations	
Expression	Column
row8.ID_Station	ID_Station
row8.Num + " " + row8.Rue + " " + row8.Ville + " " + row8.Code_postal	Adresse
row8.Tel	Tel
row8.Contact_Mail	Contact_Mail

Ici l'utilisation de TUniqRow_1 permet de filtrer les doublons qui auraient été générés par la jointure. Nous avons constaté que cela ne les élimine pas tous cela étant dû à un problème de conformité au format des attributs adresse, mail et tel.

Nous proposerons un job Talend d'évaluation et d'amélioration de cet aspect de la qualité dans la section suivante.

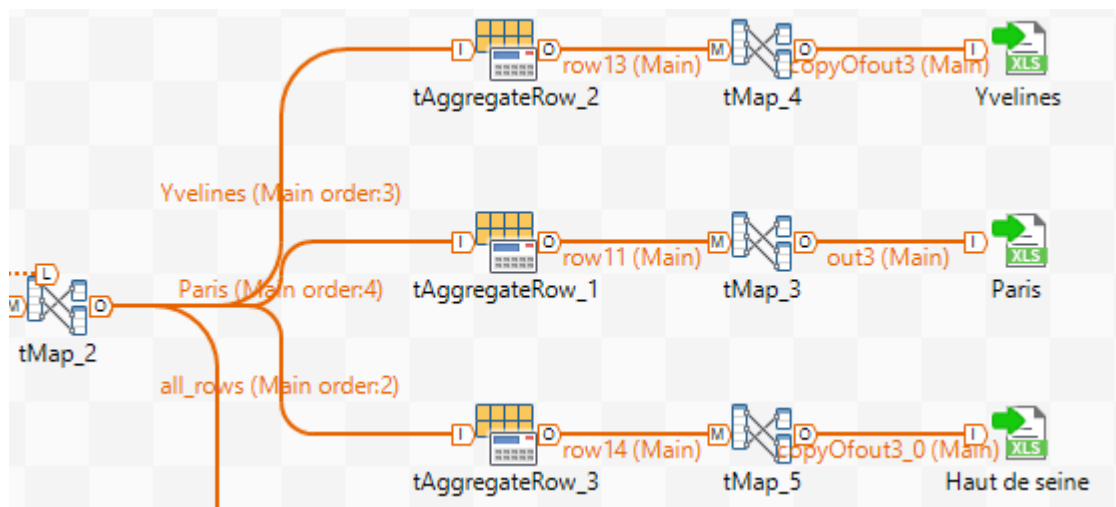


- Création des tables Paris, Yvelines, Hauts-de-Seine :


Nous effectuons un filtre sur le code postal des stations et la localisation des capteurs mobiles avec l'opérateur tMap.

Yvelines	
row8.Code_postal.startsWith("78")	
Expression	Column
row8.ID_Polluant	ID_Polluant
row21.Date	Date
row21.ID_Capteur	id_capteur
row21.Localisation	localisation
row21.Taux_releve	Taux_capteur
row21.Seuil_tolere	Seuil_tolere
row8.ID_Station	ID_Station
row8.Date	Date_station
row8.Taux_releve	Taux_Station
row8.Code_postal	Code_Postal

Ces résultats sont envoyés dans un objet tAggregateRow qui compte le nombre de capteurs mobiles, des stations et la somme des taux_relevé des capteurs mobile et des stations.



Pour calculer le taux_moyen_jour pour chaque polluant dans un département donné, nous devons de nouveau effectuer un tMap.

Expression	Column
row13.ID_Polluant	 ID_Polluant
row13.NBS	NBS
row13.NBC	NBC
$(row13.SUMS+row13.SUMC)/(row13.NBC+row13.NBS)$	Taux_Moyen_Jour
row13.Seuil_tolere	Seuil_tolere

Les tables cibles sont stockées dans des fichiers Excel nommés respectivement.

II. Problèmes d'intégration :

Lors de la phase d'intégration des données, nous avons constaté des erreurs dues aux différentes opérations utilisées dans pour matérialiser nos mappings.

Nous avons listé et résolu les problèmes suivants :

- L'apparition de doublons après l'union des stations sources :

Une fois l'union des tables stations réalisée, on constate qu'une même station existe dans deux sources différentes (S1 et S2). Ce qui se répercute dans la suite des opérations (Jointure sur ID_Polluant par exemple).

Nous avons donc utilisé l'objet tUniqRow pour garder que les champs possédant une adresse, un numéro de téléphone et un email semblable, mais du de problème de cohérence au format et de complétude, cette opération ne supprime pas tous les doublons.

Nous avons donc mis en place dans notre métamodèle, une procédure pour détecter ces doublons puis les supprimer sur la base d'un algorithme de similitude.

Les détails de la procédure d'évaluation et d'amélioration sont expliqués dans la section suivante.

- Conversion de la localisation en code postal :

La position des capteurs et des stations est exprimée sous deux formats différents. Le premier sous forme de coordonnées GPS et le second sous forme d'adresse postale (Rue, Ville, Code postal).

Dans le but de joindre ensemble les données d'un capteur et d'une station se situant dans le même département, nous avons besoin d'avoir ces valeurs d'attributs sous le même format.

Nous avons retenu le format postal pour les deux types de capteurs.

Pour convertir des coordonnées GPS en code postal, nous avons implémenté une routine Java qui envoie une requête à une API en donnant en paramètre la localisation GPS, la réponse est le code postal correspondant.

```
try
{
    URL url = new URL("https://api-adresse.data.gouv.fr/reverse/?lon=" + parts[1]+"&lat="+parts[0]);
    String res = "";

    try (BufferedReader reader = new BufferedReader(new InputStreamReader(url.openStream(), "UTF-8")))
    {
        for (String line; (line = reader.readLine()) != null;)
        {
            res += (" " + line) ;
        }

        JSONObject obj = new JSONObject(res);
        System.out.println(obj.get("features").toString());
        JSONArray a = new JSONArray(obj.get("features").toString());

        JSONObject node1 = new JSONObject(a.get(0).toString());
        JSONObject node2 = new JSONObject( node1.get("properties").toString());
        String data = node2.get("postcode").toString();

        return data ;
    }
}
```

III. Métamodèle de qualité :

Dans cette section, nous allons décrire pour chaque mesure de qualité, la structure du fichier cible, pour des raisons de lisibilité et simplicité, nous avons choisis de stocker le résultat de chaque job d'évaluation relatif à chaque table dans un fichier séparé.

-Complétude :

Pour chaque table sur laquelle a été évaluée la métrique de complétude, nous avons un fichier avec autant de colonnes dans le fichier que d'attributs de la table.

La ligne suivante représente la valeur de complétude au sens de la colonne.

-Conformité au format :

Pour ce qui est de la conformité au format, pour chaque table nous avons appliqué des filtres différents (email, téléphone, nom de capteur, valeurs de taux, etc).

Nous avons jugé nécessaire de générer un fichier par table contenant les tuples rejetées par le job d'évaluation.

La structure de ce fichier conserve la structure des tuples de la table source.

-Doublons :

Nous avons trouvé logique de combiner les jobs d'évaluation et d'amélioration des doublons, car séparer les tuples rejetées et les tuples uniques et les stocker dans des tables/fichiers intermédiaires nous a paru plus optimisé.

La structure des fichiers dits 'unique' et de rejet que nous avons appelé 'doublons' est conforme à la structure des tables sources sur lesquelles à été appliqué le job.

Une relecture de ces fichiers plus l'application d'une formule pour calculer le rapport montrent le taux de doublons dans une table. Ce taux est stocké dans un fichier "Pourcentage de doublons" avec comme attributs le nom de la table et la taux calcul

-Granularité :

Une fois, le job d'évaluation de l'écart dans la granularité exécuté, nous stockons les résultats obtenus pour chacune des tables sur lesquelles le job a été appliqué de la façon suivante.

La table contient les attributs suivants :

- Table 1 : Première table qui sert de comparaison.
- Colonne 1 : Colonne de la table 1 sur laquelle s'effectue le calcul.
- Table 2 : Seconde table qui sert à la comparaison.
- Colonne 2 : Colonne de la table 2 sur laquelle s'effectue le calcul.
- RapportT1_T2: rapport des mesures.

Le résultat du job d'amélioration correspond à un nouveau fichier source dans notre cas, avec des valeurs hétérogènes "ajustées".

IV. Facteur de qualité

Une fois l'intégration des données effectuée, nous avons mis en place des jobs pour l'évaluation de certains facteurs de qualité.

Nous décrirons dans cette section, comment et sur quelle table seront évalués les facteurs de qualité, ainsi que plusieurs

1. Complétude :

Les taux de complétude est le pourcentage de valeurs nulles parmi les valeurs relevées.

$$\text{TauxC} = ((\text{nbsn} / \text{nbt}) * 100)$$

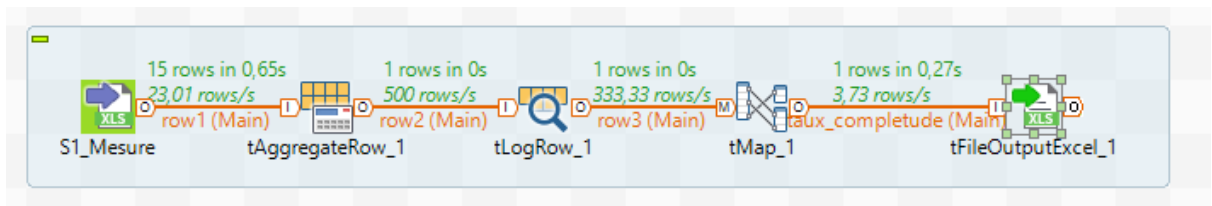
nbsn : nombre de valeurs non nulles

nbt : nombre de valeurs total de la colonne

1.1. Evaluation de la complétude :

Le calcul des taux de complétude s'effectue sur les données sources afin de repérer les valeurs nul avant les jointures avec les autres tables.

L'image suivante est un exemple permettant de calculer le taux de complétude dans le cas du "taux relevé" de la source de donnée 1 :



	A	B	C	D
1	ID_Polluant	Date	ID_Station	Taux_relevé
2	100	100	100	100
3				
4				

Dans un premier temps, nous comptons le nombre de valeurs non nulles et le nombre de valeurs totales de la colonne, représenté par tLogRow_1.

Dans un second temps nous effectuons le calcul de TauxC, ce qui nous affiche une valeur de 100 % dans l'exemple représenté par tLogRow_2.

- Source de donnée S1 :
 - Mesures :

<u>ID_Polluant</u>	<u>Date</u>	<u>ID_Station</u>	Taux_Relevé
100%	100%	100%	100%

- Stations :

<u>ID_Station</u>	Numéro	Rue	Code_postal	Ville	Téléphone	Contact_Mail
100%	71%	100%	100%	100%	100%	100%

- Source de donnée S2 :
 - Mesures :

<u>ID_Polluant</u>	<u>Date</u>	<u>ID_Station</u>	Taux_Relevé
100%	100%	100%	100%

- Stations :

<u>ID_Station</u>	Numéro	Rue	Code_postal	Ville	Téléphone	Contact_Mail
-------------------	--------	-----	-------------	-------	-----------	--------------

100%	50%	100%	100%	83%	100%	83%
------	-----	------	------	-----	------	-----

- Source de donnée S3 :
 - Polluants :

<u>ID_Polluant</u>	Désignation	Seuil toléré (µg/m3)
100%	100%	100%

- Source de donnée S4 :
 - Mesures :

<u>ID_Polluant</u>	<u>Date</u>	<u>ID_Capteur</u>	Localisation	Taux_Relevé
100%	100%	100%	87%	87%

- Source de donnée S5 :
 - Mesures :

<u>ID_Polluant</u>	<u>Date</u>	<u>ID_Capteur</u>	Localisation	Taux_Relevé
100%	100%	100%	100%	83%

Les résultats obtenus par ce job pour toutes les tables et toutes les colonnes sont stockés dans un fichier Excel.

1.2. Amélioration de la complétude :

Une fois les mesures de complétude recueillies, nous pouvons appliquer des jobs pour améliorer les données.

Les traitements à appliquer dépendent de la donnée que nous voulons compléter.

Pour les tables stations des sources, il serait pertinent de déduire le nom de la ville en fonction du code postal, car nous avons 100 % de complétude sur l'attribut code postal.

Concernant les mails et numéros de rue manquants, une solution serait de chercher dans d'autres sources de données des stations présentant les mêmes adresses et de compléter les attributs à partir des tuples "complets".

Pour ce qui est des capteurs, les mesures manquantes peuvent être déduites en prenant la moyenne pondérée par la distance de tous les capteurs à proximité du capteur possédant une mesure manquante.

Quant aux localisations manquantes, il serait pertinent de regarder si ce même capteur n'a pas émis dans les dernières minutes avant la mesure manquantes, on pourrait déduire une localisation approximative (avec un rayon d'écart basé sur le temps)

2. Conformité au format :

Pour mettre en place les mesures de conformité, nous avons pensé à utiliser des expressions régulières pour détecter les champs qui ne correspondaient pas au format que nous avons choisis, nous pouvons de ce fait calculer un taux de non-conformité et identifier les tuples et les attributs sur lesquels un traitement d'amélioration des données est nécessaire.

Le format des données est évalué sur les données sources afin de repérer les données non exploitables avant les jointures et unions avec les autres tables.

Le tableau suivant liste le format des données attendu :

Identifiant	Condition	Type
ID_Polluant	"NO2", "PM10", "PM2.5", "O3", "CO", "SO2"	String
Date	"DD-MM-AAAA HH-MM-SS"	Date
ID_Station	ID_Station > 0	Integer
Taux_Relevé	Taux_Relevé > 0	Double
Numéro	Numéro > 0	String
Rue		String
Code_Postal	"00000", 5 chiffres	String
Ville		String
Téléphone	" 00 00 00 00 00 ", 10 chiffres	String
Contact_Mail	"Ville@airparif.fr"	String
Désignation	"Dioxyde d'azote", "Particules", "Particules", "Ozone", "Monoxyde de carbone", "Dioxyde de soufre"	String
Seuil Toléré	Seuil Toléré > 0	Integer
ID_Capteur	"Can00" ou "Cairs00"	String
Localisation	"longitude, latitude"	String

- Source de donnée S1 :

- Stations : Un “Numéro” ne correspond pas au format de données des “Téléphone” de 10 chiffres, on suppose qu’il manque deux chiffres au numéro de téléphone.
(exemple : 01 40 88 88 88)

ID_Station	Numéro	Rue	Code_postal	Ville	Téléphone	Contact_Mail
6	2 bis	Quai de la Mégisserie	75001	Paris	01 44 50 75	<u>pari01@airparif.fr</u>

- Source de donnée S2 :
 - Stations : Un “Contact_Mail” ne correspond pas au format de données des “Contact_mail”, il manque le nom de domaine.
(exemple : xxxx@exemple.fr)

ID_Station	Numéro	Rue	Code_postal	Ville	Téléphone	Contact_Mail
1		Château_princeloup	78120	Sonchamp	01 34 84 41 08	<u>sonchamp@</u>

- Source de donnée S4 :
 - Mesures : On remarque 2 “Localisation” ne correspondant pas au format composé de deux nombres avec 6 chiffres après la virgule.
(exemple : 48.886933, 2.334836)

ID_Polluant	Date	ID_Capteur	Localisation	Taux_Relevé
NO2	01/10/2021 12:00	Can21	2.29	98,4
CO	02/10/2021 15:00	Can45	67	4789,67

- Source de donnée S5 :
 - Mesures : Un “ID-Capteur” est non conforme, il doit être sous la forme “CairsXXX” (exemple : Cairs123)

ID_Polluant	Date	ID_Capteur	Localisation	Taux_Relevé
NO2	08/10/2021 12:00	C	48.826501, 2.345124	0,12435

2.1. Détection des erreurs de conformité à un format :

Une fois les différentes erreurs de format détectées manuellement, nous avons choisi d’implémenter un mécanisme de détection qui repose sur l’utilisation d’une routine. Une routine est un objet java composé de différentes méthodes statiques, qui fera appelle aux expressions régulières qu’on aura définies et appliquera ces expressions sur les différents tuples des tables sources.

Ne seront renvoyés que les tuples qui ne “match” pas avec ces différentes expressions régulières.

Le code qui compare les valeurs des tuples aux expressions régulières que nous avons définies est le suivant :

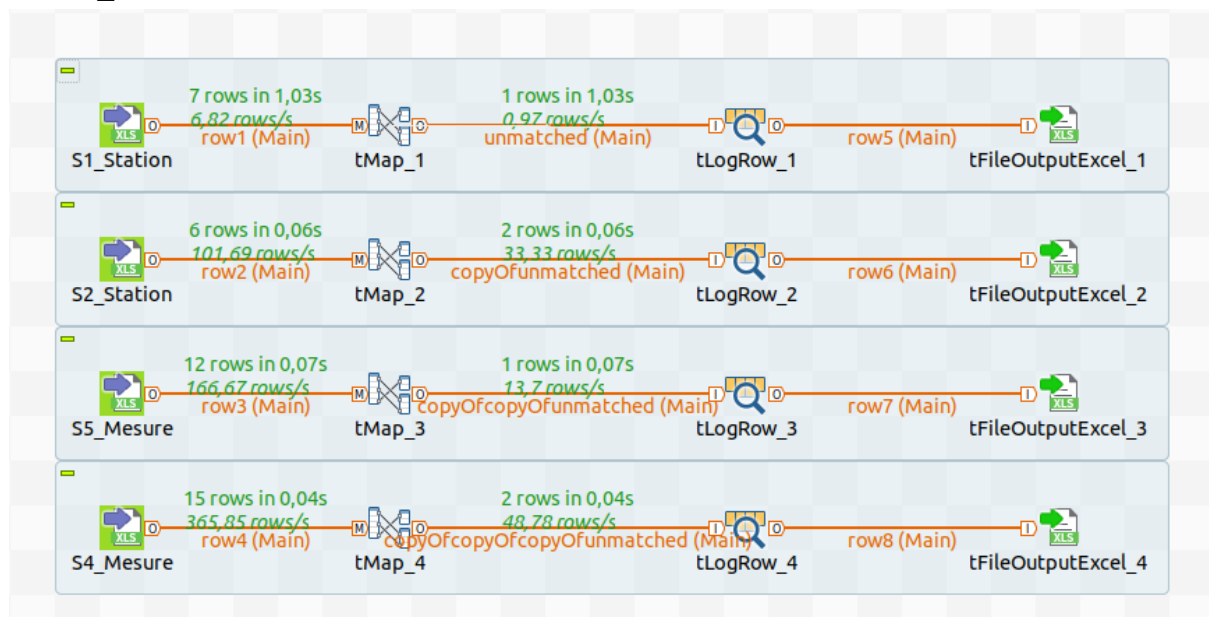
```
public class matcher {
    public static boolean matchMail(String message) {
        if (message == null) {
            return false;
        }
        if(message.toLowerCase().matches("(?:[a-z0-9!#$%&'*/=?^_`{|}~]+(?:\\.[a-z0-9!#$%&'*/=?^_`{|}~]+)*)|(?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21\\x24-\\x2c\\x2e-\\x3f\\x41-\\x4b\\x4d-\\x5f\\x61-\\x63\\x65-\\x7f\\x81-\\x83\\x85-\\x9f\\xa1-\\xa3\\xa5-\\xaf\\xb1-\\xb3\\xb5-\\xbf\\xc1-\\xc3\\xc5-\\xc7\\xc9-\\xcb\\xcd-\\xcf\\xd1-\\xd3\\xd5-\\xd7\\xd9-\\xdb\\xdd-\\xdf\\xe1-\\xe3\\xe5-\\xef\\xf1-\\xf3\\xf5-\\xf7\\xf9-\\xfb\\xfd-\\xfe\\xff])")) {
            return true;
        }
        return false;
    }

    public static boolean matchPhone(String tel) {
        if(tel.replaceAll(" ", "").length() < 10 || tel.replaceAll(" ", "").length() >= 12 ) return false;
        return true;
    }

    public static boolean matchIDCapteur(String idcapteur) {
        if (idcapteur == null) {
            return false;
        }
        if(idcapteur.toLowerCase().matches("[a-z]+[0-9]+")) return true;
        return false;
    }

    public static boolean matchlocalisation(String loc) {
        if(loc != null) {
            if (!loc.contains(",")) {
                return true;
            }
        }
        return false;
    }
}
```

Ces tuples non conformes seront insérés dans des fichiers Excel préfixés par “format_nomtable”.



À l'exécution du job nous obtenons des résultats conformes aux observations faites “manuellement”, comme le montre le screen ci-dessous :

tLogRow_1						
ID_Station	Num	Rue	Ville	Code_postal	Tel	Contact_Mail
6	2 bis	Quai de la Mégisserie	75001	Paris	01 44 50 75	pari01@airparif.fr

tLogRow_2						
ID_Station	Num	Rue	Ville	Code_postal	Tel	Contact_Mail
1	null	Château princeloup	78120	Sonchamp	01 34 84 41 08	sonchamp@
6	null	Parvis de la Défense	92800	Puteaux	01 46 92 92 92	null

tLogRow_3				
ID_Polluant	Date	Localisation	ID_Capteur	Taux_releve
NO2	08-10-2021	48.826501, 2.345124	C	0.12435

tLogRow_4				
ID_Polluant	Date	Localisation	ID_Capteur	Taux_releve
NO2	01-10-2021	2.29	Can21	98.4
CO	02-10-2021	67	Can45	4789.67

Dans l'ordre, nous avons les stations dans S1 et S2 qui ont un email ou un numéro de téléphone non conforme, puis les mesures pour lesquelles la localisation et l'ID du capteur ne sont pas conformes.

3. Doublons :

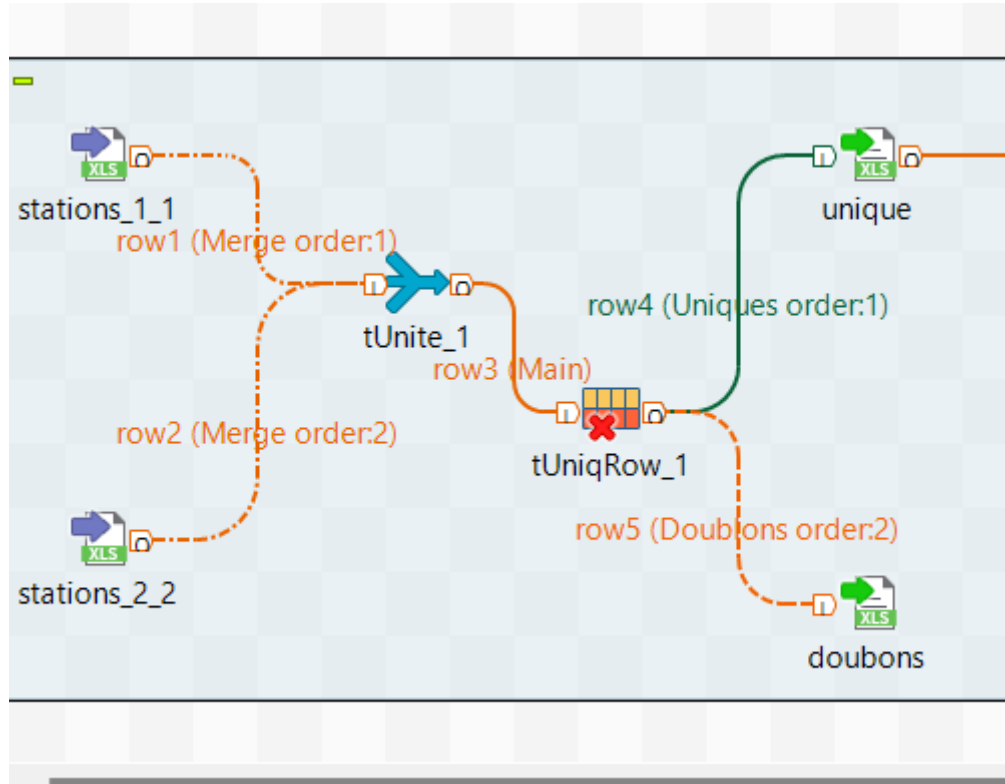
Nous appliquons le job d'évaluation de la quantité de doublons sur les tables sources :Stations S1 et Station S2 car nous ne voulons pas que l'union des deux tables nécessaire à matérialiser le mapping contient des doublons.

Nous générons pour cela une métrique qui correspond au pourcentage de doublons sur les deux tables, métrique calculée sur la base de la colonne numéro de téléphone et code postal.

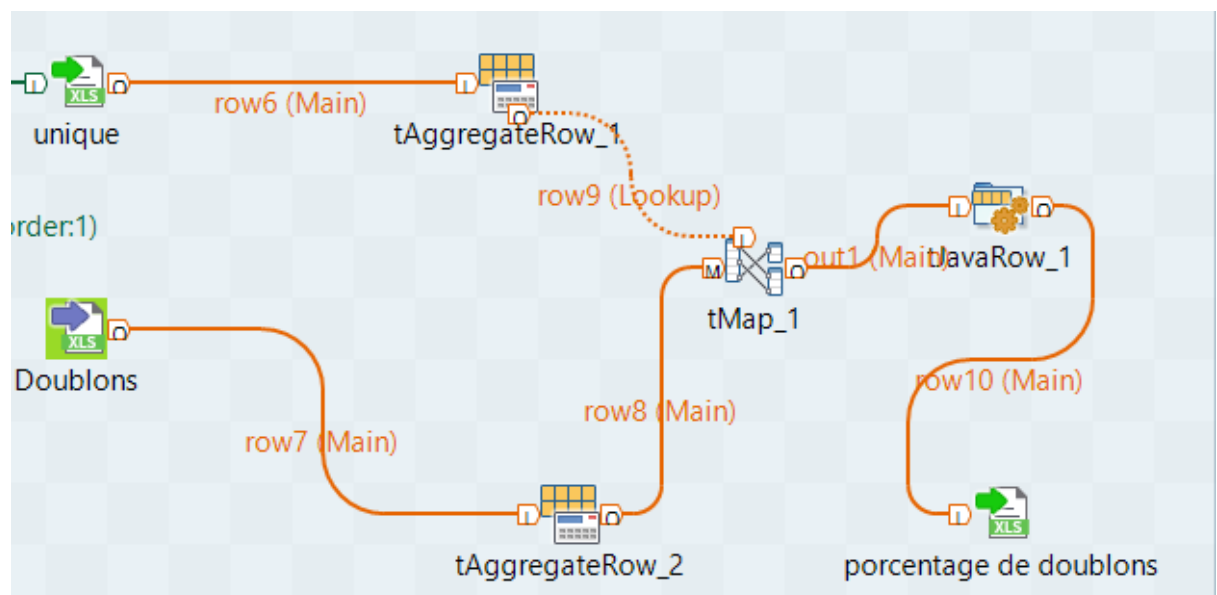
Dans le cas de notre jeu de données, il est plus intéressant d'effectuer les suppressions des doublons sur les données source après l'Union, c'est-à-dire dans la table intermédiaire tmp. Par ailleurs, il est conseillé d'effectuer la suppression des doublons dès les fichiers sources afin de minimiser les coûts.

- Stations :

Le critère d'évaluation des boulons est fait sur les attributs numéro de rue et code postal, que l'on suppose uniques pour chaque station différente. Nous pouvons ensuite penser à une méthode pour calculer la similarité entre deux chaînes de caractère et détecter au doublon au-dessus de ce seuil.



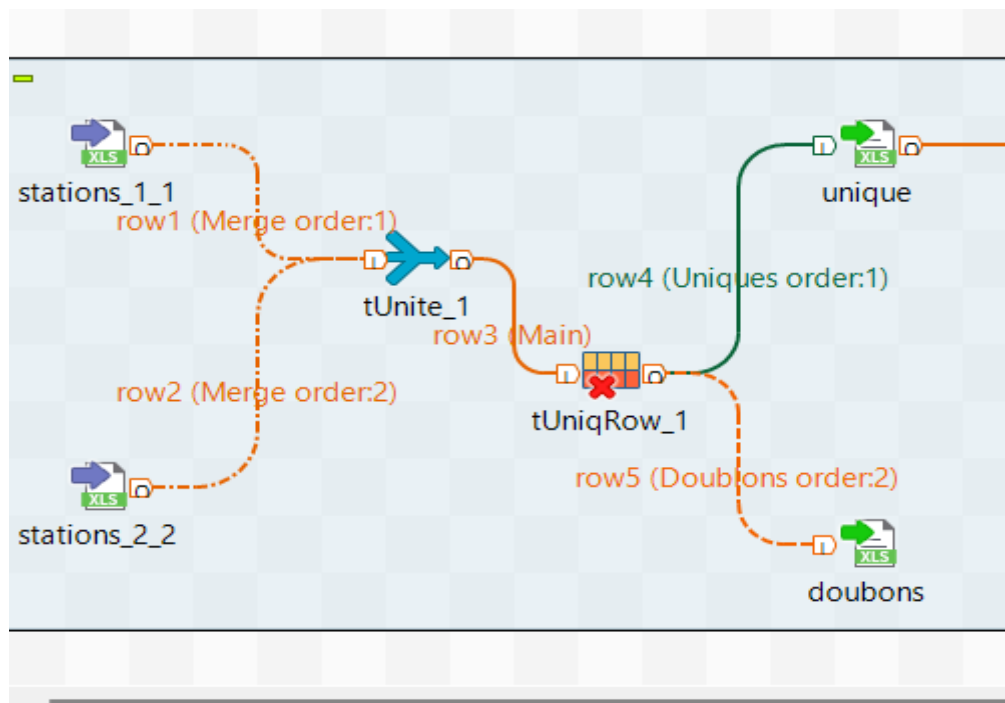
Le composant tUniqRow compare des entrées et trie les entrées en doublon du flux d'entrée.



on compte les id des uniques et les id des doublons, si les count de doublons est plus grand que 0 alors il fait $\text{count_doublons} / \text{totale}$, et on le stocke dans une table avec le critère et le table étudié et la date de résultat.

```
if(out1.CountDoublons>0)
{
    row10.pourcentage = ((double)out1.CountDoublons / (out1.CountDoublons +
out1.CountUnique))*100;
    row10.critere = "Doublons";
    row10.table = "Stations";
    row10.date = TalendDate.getCurrentDate();
}
```

3.1 Amélioration de doublons :



L'UNION des deux fichiers stations_1 et stations_2 donne une table qui contient des doublons, le composant tUniqRow_1 compare des entrées et trie les entrées en doublon et en unique, on stocke les résultats d'unique des données distincts dans un fichier xlsx.

4. Hétérogénéité ou granularité :

L'hétérogénéité des données sont les éléments de nature différente et présentent des différences de structure.

La granularité des données définit un ordre de grandeur équivalent aux mêmes types de donnée.

Certaines données peuvent être d'unité différente, l'objectif est donc d'avoir une unité commune pour les données similaires dans les différentes sources de données.

L'hétérogénéité est calculée sur les données sources pour simplifier par la suite les calculs des données cibles.

4.1. Détection des erreurs de granularité

Pour détecter l'hétérogénéité des données, nous comparons la moyenne des tuples des colonnes "Taux_Relevé" entre les Mesures de station fixe puis la moyenne pour les capteurs des sources de données.

Plus précisément nous divisons la moyenne du "Taux_Relevé" de la source de donnée 1 par la moyenne du "Taux_Relevé" de la source de donnée 2.

De même entre la source de données 4 et 5.

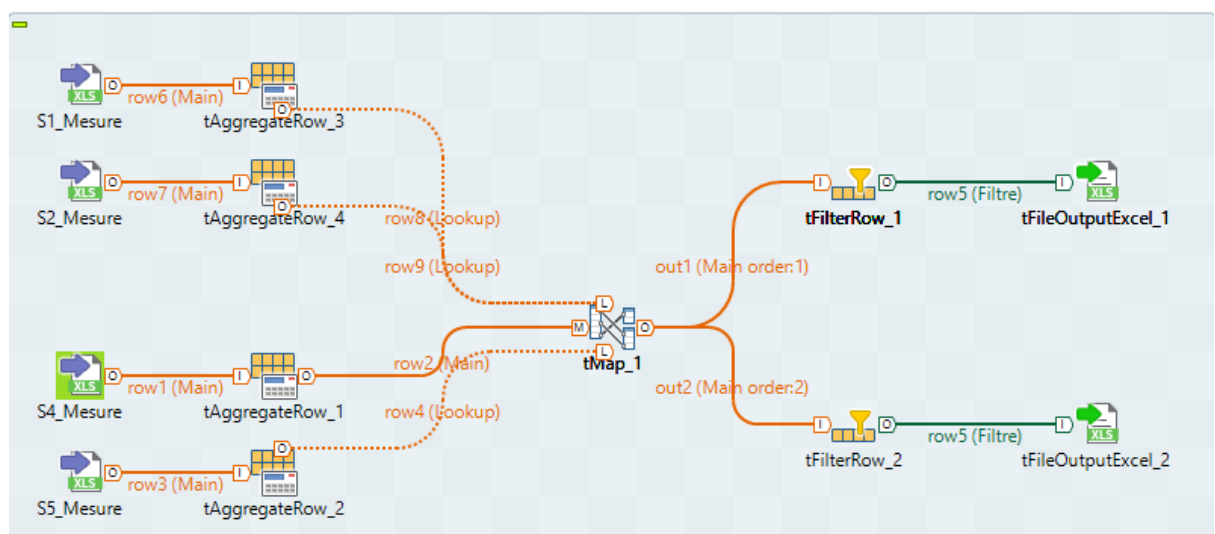
Pour respecter la règle d'hétérogénéité nous avons fait les calculs suivants :

Le calcul se fait par rapport aux valeurs des taux relevés de la source 4.

On met à l'échelle les taux de la source 5 par rapport à la moyenne des taux relevés de la source 4.

Moyenne (Source 4) = 1058, tendit que la moyenne de la source 5 est inférieure à 2.

L'image suivante est la procédure d'évaluation d'ordre de grandeur entre les sources



Dans un premier temps grâce à tAggregateRow et tMap on calcule le rapport entre la moyenne des taux relevés de la source 1 et 2, qui est de 25.

A	B	C	D	E
Table1	MoyenneT1	Table2	MoyenneT2	RapportT1_T2
Mesure1	51702,5333333	Mesure2	2052,636111	25,188358059893
Mesure1	51702,5333333	Mesure2	2052,636111	25,188358059893

Dans un second temps, on calcule le rapport entre la moyenne des taux relevés de la source 4 et 5, qui est de 500.

A	B	C	D	E
Table1	MoyenneT1	Table2	MoyenneT2	RapportT1_T2
Mesure4	1058,04307692	Mesure5	1,940116	545,3504207599
Mesure4	1058,04307692	Mesure5	1,940116	545,3504207599

On autorise une différence du taux de 100.

tFilterRow est un composant qui filtre des lignes d'entrée en définissant une ou plusieurs conditions sur les colonnes sélectionnées.

Pour les capteurs mobilité, il est 50 fois supérieure au seuil autorisé ($500 < 100$), le modèle renvoie une valeur de sortie.

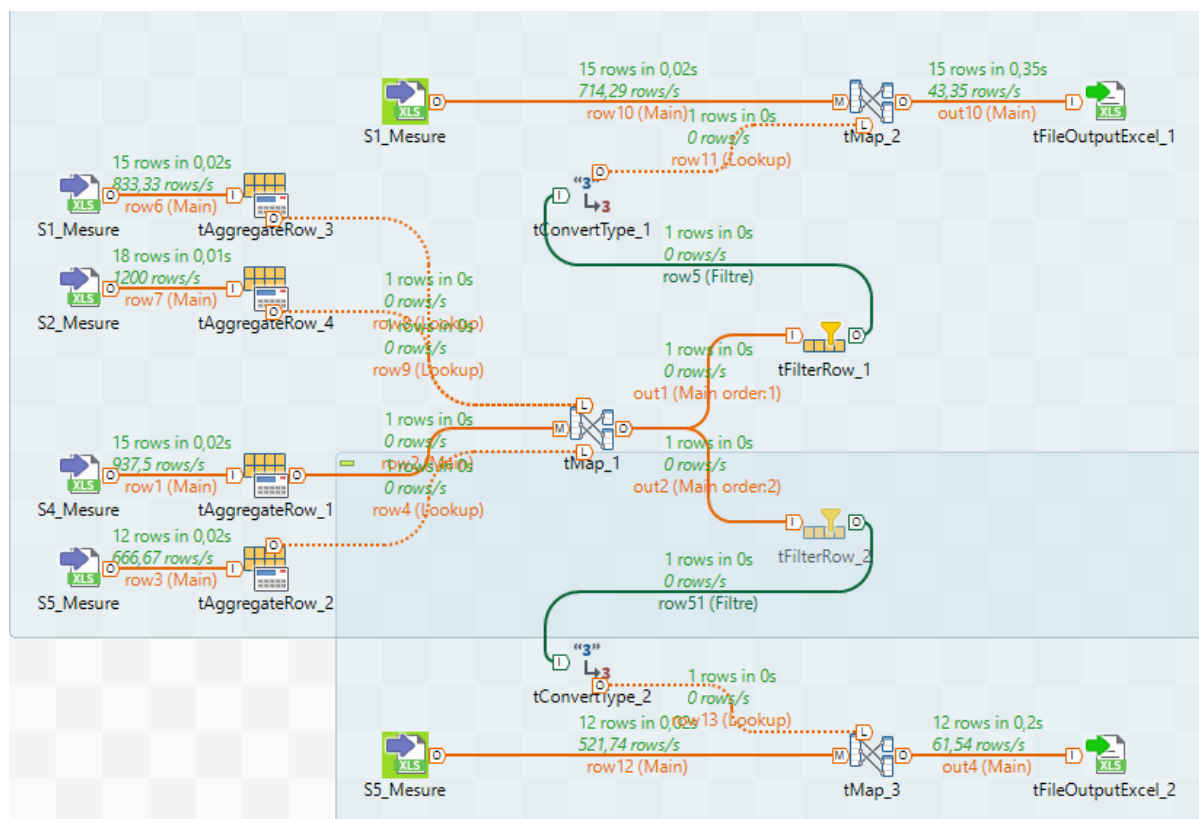
Cependant, pour les capteurs des stations statiques, $25 < 100$, le modèle ne renvoie aucune valeur de sortie.

4.2. Amélioration de la granularité

Pour améliorer la granularité des données nous devons mettre à la même échelle les données source. En effet, après analyse des données, les sources de donnée 4 et 5 n'ont pas une unité commune pour la colonne du "Taux_Relevé".

Suite à la détection des données nous avons décidé d'ajouter une procédure d'amélioration :

Pour cela nous multiplions les tuples de la colonne "Taux_Relevé" de la source 5 par le taux calculé précédemment. Ce qui donne le schéma suivant :



Pour multiplier la colonne “Taux_Relevé” de la source de donnée 5, nous devons convertir le taux en sortie de tFilterRow, car il est considéré comme une colonne. Nous devons le changer en une constante pour la multiplication que nous réalisons dans tMap_3 entre les tuples de la colonne “Taux_Relevé” de la source 5 et le taux calculé.

Les nouvelles données de la source 5 sont stocké dans un fichier Excel, affiché ci-dessous.

	A	B	C	D	E
	ID_Polluant	Date	ID_Capteur	Localisation	Taux_releve
	NO2	02-10-2021	Cairs21	48.830952, 2.331280	3221,93028585
	NO2	03-10-2021	Cairs35	48.827336, 2.327160	127,9664762313
	NO2	03-10-2021	Cairs44	48.831178, 2.318920	0
	O3	02-10-2021	Cairs22	48.837013, 2.239375	47,44548660611
	NO2	07-10-2021	Cairs89	48.825980, 2.343639	18,86912455829
	NO2	07-10-2021	Cairs123	48.831304, 2.320748	13,02842155195
	NO2	08-10-2021	C	48.826501, 2.345124	67,8143248215
	NO2	09-10-2021	Cairs4	48.821372, 2.345562	4416,247707314
1	PM10	01-10-2021	Cairs19	48.844595, 2.234076	54,53504207599
1	CO	12-10-2021	Cairs45	48.883668, 2.278408	0
2	PM10	02-10-2021	Cairs32	48.833380, 2.359467	2612,048549801